**CEPH OBJECT GATEWAY (or) RGW ( Rados Gateway )**

* Ceph Object Gateway is running on Civetweb (embedded into the ceph-radosgw daemon) instead of Apache and FastCGI. Using Civetweb simplifies the Ceph Object Gateway installation and configuration.
* To run the Ceph Object Gateway service, you should have a running Ceph storage cluster, and the gateway host should have access to the public network.

ceph-deploy install –rgw mon1

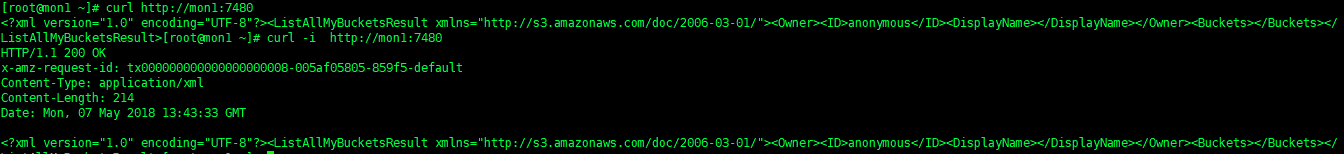
* Here we are taken the gateway node as monitor node

ceph-deploy admin mon1

* And also deployed the admin configuration in Monitor node.
* Now we create a interface of the Node

ceph-deploy rgw create mon1

* Once the instance is created and running on <http://mon1:7480>
* Now check and output of the URL given below:



* Then push all config in all nodes

ceph-deploy --overwrite-conf config push mon1 osd1 osd2 osd3 osd4 osd5

* Restart the services

sudo systemctl restart ceph-radosgw.target

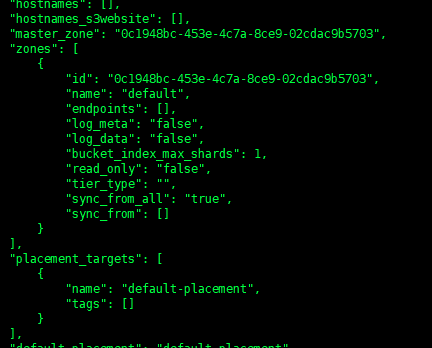
**Configure bucket sharding**

* A Ceph Object Gateway stores bucket index data in the index\_pool, which defaults to .rgw.buckets.index. Sometimes users like to put many objects (hundreds of thousands to millions of objects) in a single bucket.
* If you do not use the gateway administration interface to set quotas for the maximum number of objects per bucket, the bucket index can suffer significant performance degradation when users place large numbers of objects into a bucket.

**Export the Zone Json file** :

radosgw-admin zonegroup get > zonegroup.json

* Change the value of bucket\_index\_max\_shards value 0 to 1



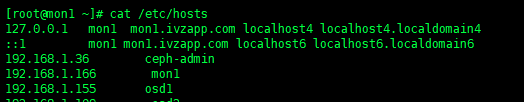
* Next set the configuration in the zonegroup

radosgw-admin zonegroup set < zonegroup.json

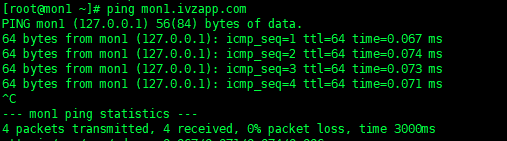
* Once you have updated your zonegroup, update and commit the period.

radosgw-admin period update –commit

* Next We added the FQDN in DNS configuration for public network access



* Ping status on configured DNS :



**Create a RADOSGW User for S3 Access:**

* We have two methods S3 and SWIFT access, Now we creating S3 access only.

sudo radosgw-admin user create --uid="vasanth" --display-name="vasanth"

**output:**

{

"user\_id": "vasanth",

"display\_name": "vasanth",

"email": "",

"suspended": 0,

"max\_buckets": 1000,

"auid": 0,

"subusers": [],

"keys": [

{

"user": "vasanth",

"access\_key": "8EB3289U2KUO56IUL805",

"secret\_key": "Dj0nt4JQngGQgDAlOqhZsRa0QH1x2pobP3k63sus"

}

],

"swift\_keys": [],

"caps": [],

"op\_mask": "read, write, delete",

"default\_placement": "",

"placement\_tags": [],

"bucket\_quota": {

"enabled": false,

"check\_on\_raw": false,

"max\_size": -1,

"max\_size\_kb": 0,

"max\_objects": -1

},

"user\_quota": {

"enabled": false,

"check\_on\_raw": false,

"max\_size": -1,

"max\_size\_kb": 0,

"max\_objects": -1

},

"temp\_url\_keys": [],

"type": "rgw"

}

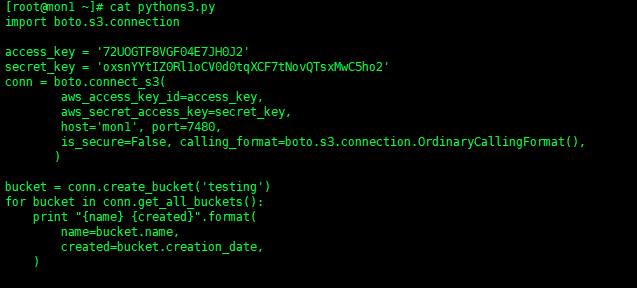
#### TEST S3 ACCESS VERIFICATION:

* You will need to install the **python-boto** package:

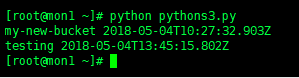
yum install **python-boto**

* We are creating a bucket the name of testing and before created “access and secret key “ placed on the script for authentication purpose.

**cat python3.py**

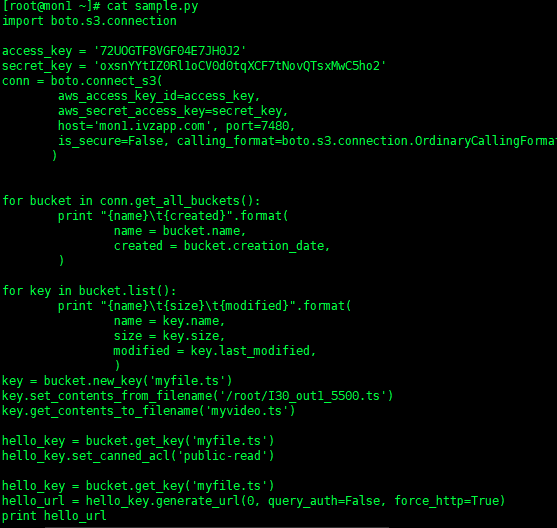


* It had my\_bucket already for my testing purpose created early.

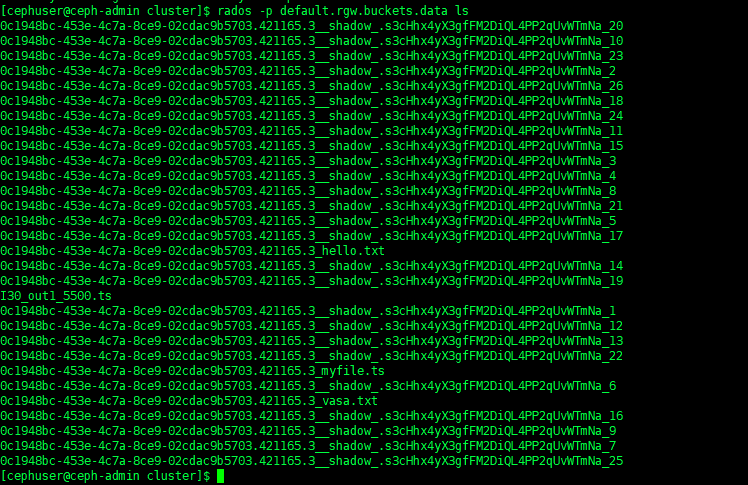


* We will put the video file on my testing bucket to added via python script

cat sample.py



* In this script , We upload a local content video file to my bucket S3 instance and We change the file name and uploaded a new name with “myfile.ts” ,We have two access given in the logic now, We are using given URL generate using “public\_read” access only <http://mon1.ivzapp.com:7480/testing/myfile.ts>
* Output of the data placed in default.rgw.buckets.data:



* Accessing the Ceph Data via http directly :

